

Appl. No. 10/041,033  
Amdt. dated July 20, 2006  
Reply to Office Action of April 21, 2006

PATENT

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

Claims 1-37. (Canceled)

38. (Currently Amended) A system for efficiently performing memory intensive computations, the system comprising:

a data cache located in memory of the system, ~~wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database, wherein the data cache [[is]] being coupled to a first set of data stored in a database;~~ and a second set of data stored in memory of the system, wherein the data cache is configured to perform a scan operation on at least a portion of the first set of data and perform a first update operation on the second set of data with changes that have occurred in the first set of data;

an engine manager coupled to the data cache and configured to instruct the data cache to perform the first update operation; and

a solver coupled to the data cache and configured to perform one or more computations [[on]] using the updated second set of data stored in the data cache memory to determine a first solution, the updated second set of data including the changes that have occurred in the first set of data, the updated second set of data being stored in memory facilitating faster computations than if the computations utilized the first set of data stored in the database,

wherein the engine manager is configured to determine if the first set of data has changed since the last first update operation, wherein if the first set of data has changed, the engine manager is configured to perform a second update operation on the second set of data in the data cache with the changes to the first set of data since the last first update operation,

wherein the solver is configured to re-perform the one or more computations [[on]] using the updated second set of data stored in the data cache memory, including the

Appl. No. 10/041,033  
Amdt. dated July 20, 2006  
Reply to Office Action of April 21, 2006

PATENT

changes that have occurred in the first set of data since the last first update operation, when the second update to the second data set occurs during performance of the one or more computations, the re-performance allowing a second solution to be determined for the one or more computations, and transmitted to the first data set, using any changes to the second data set during the performing of said one or more computations, the re-performance further causing the first solution to be discarded without first being transmitted to the first data set.

39. (Previously Presented) The system of claim 38, further comprising an application specific plug-in coupled to the solver and configured to direct the solver to perform the computations on the second set of data.

40. (Previously Presented) The system of claim 38, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data.

41. (Previously Presented) The system of claim 38, wherein the system is configured to update the second set of data with changes to the first set of data in a near-real-time fashion.

42. (Previously Presented) The system of claim 38, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data that meet a given condition.

43. (Previously Presented) The system of claim 38, wherein the data cache coupled to a first set of data and a second set of data is coupled in a bidirectional fashion.

44. (Previously Presented) The system of claim 38, wherein the first set of data comprises metadata and application data.

45. (Previously Presented) The system of claim 38, wherein the solver comprises a generic algorithms module.

Appl. No. 10/041,033  
Amdt. dated July 20, 2006  
Reply to Office Action of April 21, 2006

PATENT

46. (Previously Presented) The system of claim 38, wherein the computations solve problems encountered in business applications.

47. (New) A method for efficiently performing memory intensive computations, the method comprising the steps of:

providing a data cache located in memory of a system, the data cache being coupled to a first data set stored in a database and a second data set stored in memory of the system, at least a portion of the first and second data sets storing common data;

performing a first update operation on the second data set to reflect changes that have occurred in the first data set;

performing at least one computation using the updated second data set stored in memory to determine a first solution;

performing a second update operation on the second data set to reflect any additional changes that have occurred in the first data set since the first update operation;

re-performing the at least one computation using the updated second data set, reflecting the additional changes, to produce a second solution when the second update to the second data set occurs during performance of the at least one computation, in order to include the additional changes to the second data set in the production of the second solution; and

discarding the first solution without first transmitting the first solution to the first data set.

48. (New) A method according to claim 47, further comprising:

determining whether the first data set has changed since the first update operation

49. (New) A method according to claim 47, wherein:

said determining step includes performing a scan operation on at least a portion of the first data set.

50. (New) A method according to claim 47, further comprising:

Appl. No. 10/041,033  
Amdt. dated July 20, 2006  
Reply to Office Action of April 21, 2006

PATENT

instructing the data cache to perform the second update operation upon determination that the first data set has changed since the first update operation.

51. (New) A system for efficiently performing memory intensive computations, the system comprising:

a data caching module coupled to a first data set stored in a database and a second data set stored in memory of the system, the data caching module configured to perform a first update operation on the second data set stored in memory with any changes that have occurred in the first data set; and

a computation module operable to perform at least one computation using the updated second set of data stored in memory to determine a first solution, the updated second set of data including any changes that have occurred in the first set of data,

wherein the data caching module is further configured to perform a second update on the second data set stored in memory to reflect any additional changes to the first set of data since the first update operation, and

wherein the computation module is further operable to discard the first solution and re-perform the at least one computation using the updated second data set stored in memory, reflecting any additional changes that have occurred since the first update operation, to determine a second solution when the second update to the second data set occurs during performance of the at least one computation, in order to include any additional changes to the second data set in the determination of the second solution.